cryptography

MDPI

Article

A Practical Performance Benchmark of Post-Quantum Cryptography Across Heterogeneous Computing Environments

Maryam Abbasi [1], Filipe Cardoso [2], Paulo Váz [3], José Silva [3] and Pedro Martins [3,*]

[1] Applied Research Institute, Polytechnic of Coimbra, 3045 Coimbra, Portugal; maryam.abbasi@ipc.pt
[2] Polytechnic Institute of Santarém, Escola Superior de Gestão e Tecnologia de Santarém, 2001 Santarém, Portugal; filipe.cardoso@esg.ipsantarem.pt
[3] Research Center in Digital Services, Polytechnic of Viseu, 3054 Viseu, Portugal; paulovaz@estgv.ipv.pt (P.V.); jsilva@estgv.ipv.pt (J.S.)
* Correspondence: pedromom@estgv.ipv.pt

Abstract: The emergence of large-scale quantum computing presents an imminent threat to contemporary public-key cryptosystems, with quantum algorithms such as Shor's algorithm capable of efficiently breaking RSA and elliptic curve cryptography (ECC). This vulnerability has catalyzed accelerated standardization efforts for post-quantum cryptography (PQC) by the U.S. National Institute of Standards and Technology (NIST) and global security stakeholders. While theoretical security analysis of these quantum-resistant algorithms has advanced considerably, comprehensive real-world performance benchmarks spanning diverse computing environments—from high-performance cloud infrastructure to severely resource-constrained IoT devices—remain insufficient for informed deployment planning. This paper presents the most extensive cross-platform empirical evaluation to date of NIST-selected PQC algorithms, including CRYSTALS-Kyber and NTRU for key encapsulation mechanisms (KEMs), alongside BIKE as a code-based alternative, and CRYSTALS-Dilithium and Falcon for digital signatures. Our systematic benchmarking framework measures computational latency, memory utilization, key sizes, and protocol overhead across multiple security levels (NIST Levels 1, 3, and 5) in three distinct hardware environments and various network conditions. Results demonstrate that contemporary server architectures can implement these algorithms with negligible performance impact (<5% additional latency), making immediate adoption feasible for cloud services. In contrast, resource-constrained devices experience more significant overhead, with computational demands varying by up to 12× between algorithms at equivalent security levels, highlighting the importance of algorithm selection for edge deployments. Beyond standalone algorithm performance, we analyze integration challenges within existing security protocols, revealing that naive implementation of PQC in TLS 1.3 can increase handshake size by up to 7× compared to classical approaches. To address this, we propose and evaluate three optimization strategies that reduce bandwidth requirements by 40–60% without compromising security guarantees. Our investigation further encompasses memory-constrained implementation techniques, side-channel resistance measures, and hybrid classical-quantum approaches for transitional deployments. Based on these comprehensive findings, we present a risk-based migration framework and algorithm selection guidelines tailored to specific use cases, including financial transactions, secure firmware updates, vehicle-to-infrastructure communications, and IoT fleet management. This practical roadmap enables organizations to strategically prioritize systems for quantum-resistant upgrades based on data sensitivity, resource constraints, and technical feasibility. Our results conclusively demonstrate that PQC is deployment-ready for most applications, provided that implementations are carefully optimized for the specific performance characteristics and security requirements of target environments. We also identify several remaining research challenges for the

check for updates

Academic Editor: Josef Pieprzyk

Received: 25 March 2025
Revised: 5 May 2025
Accepted: 8 May 2025
Published: 21 May 2025

Citation: Abbasi, M.; Cardoso, F.; Váz, P.; Silva, J.; Martins, P. A Practical Performance Benchmark of Post-Quantum Cryptography Across Heterogeneous Computing Environments. *Cryptography* **2025**, *9*, 32. https://doi.org/10.3390/cryptography9020032

Copyright: © 2025 by the authors. Licensee MDPI, Basel, Switzerland. This article is an open access article distributed under the terms and conditions of the Creative Commons Attribution (CC BY) license (https://creativecommons.org/licenses/by/4.0/).

community, including further optimization for ultra-constrained devices, standardization of hybrid schemes, and hardware acceleration opportunities.

Keywords: post-quantum cryptography; quantum-resistant algorithms; lattice-based cryptography; PQC performance benchmarks; CRYSTALS-Kyber; NTRU; BIKE; resource-constrained computing; heterogeneous computing environments; TLS protocol integration; energy-efficient cryptography; NIST standardization

1. Introduction

The emergence of large-scale quantum computing represents a significant threat to modern cryptographic infrastructure. Classical public-key cryptosystems—including RSA, Diffie-Hellman, and elliptic curve cryptography (ECC)—derive their security from the computational intractability of integer factorization and discrete logarithm problems. However, quantum algorithms such as Shor's algorithm [1] can solve these problems in polynomial time, rendering current public-key infrastructure (PKI) fundamentally vulnerable. Recent advancements suggest that within the next decade, quantum computing hardware may reach the threshold required to break 2048-bit RSA and comparable ECC implementations [2,3]. This "harvest now, decrypt later" threat model—where adversaries store encrypted communications today for future quantum-enabled decryption—has mobilized governments, standards bodies, and industry stakeholders to accelerate development and deployment of post-quantum cryptography (PQC).

In response to this emerging vulnerability, the U.S. National Institute of Standards and Technology (NIST) initiated a comprehensive standardization process for quantum-resistant cryptographic algorithms in 2016 [4,5]. As of 2023, this process has yielded several candidate algorithms for standardization, with CRYSTALS-Kyber selected as the primary key encapsulation mechanism (KEM), and CRYSTALS-Dilithium, Falcon, and SPHINCS+ designated for digital signatures. Parallel efforts by the European Telecommunications Standards Institute (ETSI), the Internet Engineering Task Force (IETF), and other international bodies have further emphasized the urgent need to transition critical digital infrastructure to quantum-resistant cryptography.

Despite these significant standardization efforts, a critical gap persists between theoretical cryptographic security and real-world implementation readiness. While numerous studies have examined the mathematical foundations and security properties of PQC algorithms, comprehensive performance evaluations across heterogeneous computing environments remain limited. Existing benchmarks typically focus on either theoretical complexity analyses or isolated performance measurements on specialized platforms [6,7]. This narrow approach fails to address the diverse nature of modern computing ecosystems, where cryptographic protocols must operate efficiently across environments ranging from high-performance cloud servers to severely resource-constrained IoT devices. Moreover, the practical implications of integrating PQC into established security protocols like TLS, SSH, and IPsec—particularly regarding bandwidth overhead, latency, and backward compatibility—remain inadequately characterized.

To address these critical knowledge gaps, our research pursues three primary objectives:

1. Comprehensive Cross-Platform Performance Analysis: We conduct extensive empirical evaluations of five leading PQC algorithms: CRYSTALS-Kyber, NTRU, and BIKE for key encapsulation mechanisms (KEMs), alongside CRYSTALS-Dilithium and Falcon for digital signatures. Our testing encompasses performance metrics such as computational latency, memory utilization, energy consumption, and cryptographic

material sizes across three distinct hardware profiles ranging from high-performance servers to resource-constrained embedded devices, providing insights directly applicable to diverse deployment scenarios.

2. Protocol Integration and Network Impact Assessment: We systematically analyze the practical challenges of integrating PQC into existing cryptographic protocols, with particular emphasis on TLS 1.3 handshakes. This includes quantifying handshake latency increases, packet fragmentation effects under various network conditions, and bandwidth consumption patterns. We also evaluate hybrid classical-quantum approaches that maintain compatibility during transition periods and measure performance scaling under high-concurrency loads.
3. Deployment Strategy Framework: Based on our empirical findings, we develop a risk-based migration framework and algorithm selection guidelines tailored to specific deployment environments and use cases. These recommendations account for varying security requirements, performance constraints, and operational considerations across sectors ranging from financial services to IoT infrastructure.

Our work makes several significant contributions to the field. First, we present the most extensive cross-platform benchmarking of NIST-selected PQC algorithms to date, covering both lattice-based and code-based approaches across multiple security levels (NIST Levels 1, 3, and 5). Second, we identify and quantify critical performance trade-offs that inform algorithm selection for resource-constrained environments, including specific optimizations that yield up to 40% performance improvement on targeted platforms. Third, we evaluate practical protocol integration challenges, revealing that naive PQC implementation in TLS 1.3 can increase handshake size by up to 7× compared to classical approaches, and demonstrate mitigation strategies that significantly reduce this overhead.

By bridging the gap between cryptographic theory and systems engineering practice, our findings provide actionable guidance for security architects, protocol designers, and policymakers navigating the global transition toward quantum-resistant infrastructure. As quantum computing continues its rapid advancement, the insights presented in this paper help ensure that cryptographic protections evolve in parallel with emerging threats, preserving the confidentiality and integrity of sensitive communications in the post-quantum era.

The remainder of this paper is organized as follows: Section 2 provides a comprehensive overview of quantum computing threats and the current state of NIST's PQC standardization process, including a detailed survey of related work. Section 3 presents our algorithm selection criteria, testing environments, and experimental methodology. Section 4 details our benchmarking results across various platforms and operational scenarios. Section 5 analyzes these findings in the context of real-world deployment considerations, offering recommendations for algorithm selection and implementation strategies. This section also summarizes our key contributions and outlines directions for future research in this rapidly evolving field.

2. Background and Related Work

The looming threat of large-scale quantum computing has intensified research into post-quantum cryptography (PQC). This section provides a comprehensive overview of quantum vulnerabilities affecting classical cryptography, examines leading PQC candidates in the standardization pipeline, and surveys existing benchmark studies—identifying the gaps our work aims to address.

2.1. Quantum Attacks and Their Impact on Classical Cryptography

Peter Shor's groundbreaking algorithm demonstrated that integer factorization and discrete logarithms can be solved in polynomial time on a fault-tolerant quantum computer [1]. This discovery directly endangers RSA, Diffie-Hellman (DH), and elliptic curve cryptography (ECC), all of which rely on the presumed intractability of these problems in the classical model. Recent studies have refined Shor's approach and provided more concrete resource estimates, with Gidney and Ekerå [2] suggesting that factoring 2048-bit RSA could become feasible within the next decade given aggressive quantum hardware advancements. Gheorghiu et al. [3] further analyzed the resource requirements for implementing Shor's algorithm, confirming the potential vulnerability of current cryptographic standards.

Beyond factoring-based attacks, Grover's algorithm offers a quadratic speedup for brute-forcing symmetric cryptographic keys [8], motivating the need to increase key sizes in block ciphers and hash functions. NIST's guidelines [9] recommend doubling symmetric key lengths to maintain equivalent security margins against quantum attacks. However, as Mosca [10] emphasizes, the most urgent concern lies in public-key infrastructures (PKIs), where classical algorithms could be retroactively broken once a sufficiently powerful quantum computer is built. This "store now, decrypt later" threat has accelerated exploration of quantum-resistant alternatives across both government and industry sectors [4].

2.2. NIST PQC Standardization Process and Leading Candidates

To systematically address these threats, NIST launched a multi-round standardization process in 2016, soliciting post-quantum algorithms from the global cryptography community [4]. After extensive cryptanalysis and performance evaluations, NIST announced their first selections in 2022, with further refinements in the subsequent "fourth round" [5,11].

2.2.1. CRYSTALS-Kyber

CRYSTALS-Kyber is a Key Encapsulation Mechanism (KEM) based on the module learning-with-errors (MLWE) problem [12,13]. With relatively low latency and moderate key/ciphertext sizes, it excels in TLS-like key exchange scenarios. NIST has selected Kyber as the primary recommendation for post-quantum key establishment, citing its balanced performance profile across diverse hardware environments. Subsequent optimizations by Abdulrahman et al. [14] have further improved Kyber's implementation efficiency, particularly on ARM platforms.

2.2.2. CRYSTALS-Dilithium

A sister scheme to Kyber, CRYSTALS-Dilithium implements lattice-based digital signatures using the same MLWE framework [15]. While Dilithium offers robust security and efficient signing/verification, its signature sizes (2.7 KB at Level 3) are larger than classical ECDSA. Nonetheless, it ranks among NIST's favored signature mechanisms due to its simplicity and well-understood security. Recent work by Greconici et al. [16] has demonstrated improved implementations with reduced memory requirements.

2.2.3. Falcon

Falcon adopts a more intricate lattice-based design rooted in the NTRU family and Fast Fourier Sampling [17]. It boasts compact signatures (approximately 63% smaller than Dilithium at comparable security levels) and efficient verification, but is recognized as being more challenging to implement securely against side-channel attacks. Pornin [18] has proposed constant-time implementation techniques to address some of these concerns.

2.2.4. NTRU and NTRU Prime

Originally devised in the 1990s [19], NTRU has undergone extensive refinements to defend against various lattice- and algebraic-based attacks. Modern variants such as NTRU-HRSS [20] and NTRU Prime [21] incorporate strengthened security properties and improved implementation characteristics. While not selected for standardization in NIST's initial round, NTRU remains a prominent alternative KEM for those seeking algorithmic diversity beyond Kyber.

2.2.5. BIKE and Other Code-Based Alternatives

BIKE (Bit-flipping Key Encapsulation) represents the code-based approach to post-quantum cryptography, building on quasi-cyclic moderate-density parity-check codes [22,23]. NIST has advanced BIKE to alternate candidate status, recognizing its value in providing algorithmic diversity against potential breakthroughs in lattice cryptanalysis. Other promising candidates include FrodoKEM [24], which offers conservative security at the cost of larger keys, and SPHINCS+ [25], a stateless hash-based signature scheme with strong security foundations but higher computational overhead.

2.3. Comprehensive Survey of Existing Benchmark Studies

2.3.1. Theoretical Analyses and Early Implementations

Initial evaluations of post-quantum schemes focused primarily on asymptotic complexity and microbenchmarks on commodity hardware. Alkim et al. [26] provided early performance comparisons of lattice-based KEMs, while Chen et al. [27] documented implementation considerations for standardization. Kannwischer et al. [28] established baseline performance metrics for multiple PQC candidates on high-end x86 processors. While these works provided valuable insights into cryptographic efficiency, they offered limited guidance on practical deployment scenarios involving diverse hardware constraints, network latencies, and large-scale concurrency.

2.3.2. Performance on Constrained Platforms

Multiple recent studies have addressed PQC performance in embedded or IoT environments. Chen et al. [6] conducted a comprehensive evaluation of lattice-based KEMs (Kyber, NTRU, Saber) on ARM Cortex-M microcontrollers, revealing significant overhead at higher NIST security levels. Mera et al. [29] specifically analyzed the energy implications of post-quantum algorithms on battery-powered devices, concluding that algorithm selection can substantially impact device lifespan. Similarly, Oder and Pöppelmann [30] optimized lattice arithmetic to fit resource-constrained devices, yet found that frequent key generation or re-keying could still significantly drain battery resources.

Wang et al. [31] recently explored specialized implementations for ultra-constrained environments (sub-100KB RAM), demonstrating that careful memory management can enable limited PQC functionality even on the most restricted hardware. Complementing this work, Kudinov et al. [32] examined the feasibility of hardware acceleration for specific PQC operations on IoT-class devices, finding promising energy-performance trade-offs for dedicated accelerators.

2.3.3. Enterprise, Cloud, and Network-Level Benchmarks

On the high-performance end, several studies have examined PQC integration into communication protocols and high-throughput environments. Hamburg [7] conducted detailed measurements of PQC overhead in TLS 1.3 handshakes, showing that advanced KEMs (e.g., Kyber) add only modest latency compared to classical ECDH, especially on powerful servers. Sikeridis et al. [33] extended this analysis to include network effects

and connection establishment rates under load, providing insights into enterprise-scale deployment considerations.

Hybrid approaches—where classical and post-quantum key exchange run concurrently—have been studied by Bindel et al. [34] and Kwiatkowski et al. [35], demonstrating that these transitional schemes offer defense-in-depth at a modest computational penalty. Google's early experiments with post-quantum TLS [36] highlighted practical deployment considerations and helped identify potential integration challenges before standardization.

Recent work by Crockett et al. [37] specifically examined scaling behavior of PQC handshakes under high concurrency, providing valuable insights for cloud service providers. Similarly, Drucker and Gueron [38] analyzed the impact of PQC integration on high-throughput VPN gateways, identifying potential bottlenecks and optimization opportunities.

2.3.4. Side-Channel and Hardware-Specific Considerations

Although many performance benchmarks omit side-channel aspects, recent research has highlighted that PQC schemes may be particularly vulnerable to certain classes of attacks. Fouque and Tunstall [39] demonstrated timing and power-analysis vulnerabilities in several lattice-based implementations, while Xu et al. [40] proposed countermeasures based on blinding techniques and constant-time operations.

Hardware acceleration represents another active research area, with Basu et al. [41] exploring FPGA implementations of leading PQC candidates, and Banerjee et al. [42] proposing specialized hardware extensions for polynomial arithmetic in lattice-based cryptography. These approaches show promising performance improvements but require significant design effort and may not be immediately deployable in existing systems.

2.3.5. Comparison Tables from Existing Studies

Table 1 summarizes key benchmark studies and highlights their specific focus areas, demonstrating the fragmented nature of existing research.

Table 1. Comparison of previous PQC benchmark studies and their limitations.

Study	Hardware Focus	Algorithms	Network/Protocol	Limitations
Chen et al. [6]	IoT (Cortex-M4)	Kyber, NTRU	No	Single platform
Hamburg [7]	Server	Kyber, NTRU	TLS 1.3	Limited hardware diversity
Mera et al. [29]	Battery-powered	Multiple	No	No protocol integration
Sikeridis et al. [33]	Server	Kyber, Saber	TLS 1.3	No constrained devices
Xu et al. [40]	Mixed	Lattice-based	No	Focus on side-channels only
Kwiatkowski et al. [35]	Cloud	Hybrid schemes	TLS 1.3	Limited algorithm coverage
Wang et al. [31]	Ultra-constrained	Kyber, NTRU	No	No server comparison

2.3.6. Unaddressed Gaps in Existing Research

While the body of PQC research has expanded significantly, our analysis reveals several persistent gaps:

- Cross-Platform Heterogeneity: Few studies offer a single, consistent benchmark suite that spans cloud servers, standard workstations, and truly constrained IoT boards. Most focus on either high-performance or low-power environments, but rarely both with identical methodologies.
- Updated Library Versions: Many published results rely on older liboqs or OpenSSL forks (pre-2022), whereas ongoing optimization in PQC libraries can yield significant performance gains in newer releases [43]. Our work specifically addresses this by using the latest implementations.

- Limited Algorithm Diversity: Most studies focus exclusively on lattice-based schemes, with fewer comprehensive evaluations of code-based alternatives like BIKE. Direct comparisons between different mathematical approaches are particularly valuable for security diversification strategies.
- Realistic Network Environments: Benchmarks incorporating practical network conditions (latency, packet loss, MTU restrictions) remain scarce, as do analyses of protocol-level impacts such as fragmentation and connection establishment rates at scale.
- Energy Consumption: Few studies provide detailed energy measurements across multiple algorithms and security levels, limiting insight into the operational impacts on battery-powered devices.

Our research directly addresses these gaps by providing a comprehensive, up-to-date experimental evaluation that integrates five distinct PQC algorithms (representing both lattice-based and code-based approaches) across three diverse hardware profiles. We measure not only raw performance but also memory usage, energy consumption, protocol integration overhead, and scaling characteristics in a consistent methodology. This holistic approach enables direct comparisons that can guide practitioners in selecting appropriate algorithms for specific deployment scenarios, from cloud data centers to resource-constrained IoT environments.

3. Methodology and Experimental Setup

This section presents our comprehensive approach to evaluating post-quantum cryptographic (PQC) algorithms across diverse computing environments. We detail the selected algorithms, testing platforms, measurement procedures, and mathematical foundations to ensure both scientific rigor and reproducibility of our results.

3.1. Algorithm Selection and Mathematical Foundations

To provide a representative assessment of the PQC landscape, we selected five algorithms spanning different mathematical approaches and use cases:

- CRYSTALS-Kyber: A lattice-based Key Encapsulation Mechanism (KEM) built on the Module Learning With Errors (MLWE) problem. Kyber's security relies on the difficulty of finding a secret vector s given a random matrix $A \in R_q^{m \times k}$ and vector $b = As + e$, where e is a small error vector and operations occur in a polynomial ring $R_q = \mathbb{Z}_q[X]/(X^n + 1)$ with $n = 256$ and $q = 3329$.
- NTRU: A lattice-based KEM using a different approach based on polynomial quotient rings. The NTRU problem involves finding small polynomials f and g such that $f \cdot h \equiv g \pmod{q}$ where h is public. Security is based on the difficulty of finding these small polynomials.
- BIKE (Bit Flipping Key Encapsulation): A code-based KEM using quasi-cyclic moderate-density parity-check codes. Unlike lattice-based schemes, BIKE's security is founded on the difficulty of decoding random linear codes, providing algorithmic diversity in our evaluation portfolio.
- CRYSTALS-Dilithium: A lattice-based digital signature scheme that shares its mathematical foundation with Kyber. Dilithium employs a "Fiat-Shamir with aborts" paradigm to transform an interactive proof system into a non-interactive signature scheme.
- Falcon: A lattice-based signature scheme using NTRU lattices with Fast Fourier sampling. Falcon achieves compact signatures through a sophisticated sampling technique over a structured lattice.

For each algorithm, we evaluated all parameter sets corresponding to NIST security levels 1, 3, and 5, which are calibrated to provide security comparable to AES-128, AES-192, and AES-256 respectively, even against quantum attackers. This comprehensive coverage allows us to analyze performance scaling with increasing security requirements.

The formal mathematical operations for each algorithm are summarized below:

CRYSTALS-Kyber:

- Key Generation: $(pk, sk) = (b = As + e, s)$ where A is random, s is secret, and e is a small error vector.
- Encryption: $c = (c_1, c_2) = (A^T r + e_1, t^T r + e_2 + \lfloor q/2 \rfloor \cdot m)$ where r is a random polynomial and m is the message.
- Decryption: $m = \lfloor \frac{q}{2} \rfloor^{-1} \cdot (c_2 - s^T c_1) \mod q$

NTRU:

- Key Generation: Find small polynomials $f, g \in R$ where $R = \mathbb{Z}[X]/(X^n - 1)$ and compute $h = g \cdot f^{-1} \in R_q$
- Encryption: $c = r \cdot h + m \mod q$, where r is a random small polynomial
- Decryption: $m = f \cdot c \mod q \mod p$

CRYSTALS-Dilithium:

- Key Generation: $(A, t = As + e)$, where s is the secret key
- Signing: Uses commitment-challenge-response with $y = Aw$, $c = H(A, t, w, m)$, $z = y + cs$
- Verification: Check $\|z\| < B$ and $c = H(A, t, Az - ct, m)$

Falcon:

- Key Generation: Find small (f, g) with NTRU relation and compute a Gram-Schmidt basis
- Signing: Sample a short vector s such that $s \equiv c \mod q$ using the Gram-Schmidt basis
- Verification: Check $\|s\| < B$ and $s \equiv c \mod q$

3.2. Testing Environments

To capture performance characteristics across the computing spectrum, we established three representative hardware platforms that reflect common deployment scenarios from high-performance servers to resource-constrained edge devices:

These environments represent distinct operational scenarios:

- Server E1 represents high-performance cloud or data center infrastructure where computational resources are abundant and processing power is prioritized.
- Laptop E2 represents typical end-user devices used by individuals and small businesses, with moderate performance capabilities.
- Device E3 represents resource-constrained edge devices, similar to those found in IoT deployments, where power consumption and memory constraints are significant concerns.

This diverse selection allows us to comprehensively evaluate how PQC algorithms perform across the computing spectrum and identify the most suitable algorithms for each environment. Table 2 describes the Hardware specifications of the test environments.

Table 2. Hardware specifications of the test environments.

Environment	Server E1	Laptop E2	Device E3
CPU	Intel Xeon Gold 6248R (3.0 GHz base, 4.0 GHz turbo), 8 physical cores, 16 logical cores	Intel Core i5-10210U (1.6 GHz base, 4.2 GHz turbo), 4 physical cores, 8 logical cores	ARM Cortex-A53 (1.4 GHz), 4 cores
RAM	32 GB DDR4-2933 MHz ECC memory	8 GB DDR4-2666 MHz memory	1 GB LPDDR2
Storage	NVMe SSD with 3.5 GB/s read, 2.7 GB/s write throughput	SATA SSD with 550 MB/s read, 520 MB/s write throughput	16 GB eMMC flash storage
OS	Ubuntu Server 22.04.2 LTS (64-bit)	Windows 10 Pro 21H2 (64-bit) with WSL2	Debian 11 "Bullseye" (32-bit)
Compiler	GCC 11.3.0 with -O3 -march=native -flto	GCC 10.3.0 with -O2 -march=native	GCC 10.2.1 with -O2 -mcpu=cortex-a53
Network	10 GbE interface	802.11ac Wi-Fi and Gigabit Ethernet	100 Mbps Ethernet
Monitoring	Performance counters	Process monitoring tools	External power meter for energy consumption

3.3. Software Environment and Implementation Details

To ensure reliable and consistent results, we standardized the software environment across all platforms:

- Cryptographic Libraries:
  - Open Quantum Safe (OQS) liboqs version 0.7.2 for core algorithm implementations
  - OpenSSL-OQS fork (version 1.1.1m-OQS) for TLS protocol integration testing
  - Original reference implementations from algorithm authors for verification and validation
- Runtime Environment Configuration:
  - CPU frequency governors set to "performance" on Server E1 and Laptop E2, and "ondemand" on Device E3 to reflect typical deployment configurations
  - Non-essential background services disabled to minimize interference
  - Thermal throttling monitoring enabled to detect and exclude thermally-constrained test runs
  - Large physical memory pages allocated where supported to reduce TLB misses and optimize memory access
- Compilation Settings:
  - Platform-specific optimization flags to leverage available hardware capabilities
  - Link-time optimization (LTO) on Server E1 to maximize performance
  - Position-independent code (PIC) disabled where possible to eliminate indirection overhead
  - Assembly optimizations enabled for all platforms where available

3.4. Measurement Methodology

Our experimental methodology was designed to collect statistically robust data across multiple dimensions:

3.4.1. Core Cryptographic Operations

For each algorithm and security level, we measured the fundamental operations with the following procedure:

- Key Encapsulation Mechanisms (KEMs):
  - Key generation time
  - Encapsulation (encryption) time
  - Decapsulation (decryption) time
  - Key, ciphertext, and secret sizes
- Signature Schemes:
  - Key generation time
  - Signing time
  - Verification time
  - Public key, private key, and signature sizes
- Measurement Protocol:
  - Each operation was executed 1000 times in succession
  - High-precision timers: clock_gettime(CLOCK_MONOTONIC_RAW) with nanosecond resolution
  - Process isolation to prevent cacheline interference or other systematic biases
  - Arithmetic mean, median, standard deviation, and 95th percentile values collected to account for non-normal distributions
  - Modified Z-score method applied to identify and handle statistical outliers

3.4.2. Resource Utilization Metrics

Beyond timing measurements, we captured comprehensive resource utilization:

- Memory Usage:
  - Peak resident set size (RSS) during each cryptographic operation
  - Heap allocation patterns and memory fragmentation analysis
  - Stack depth requirements for constrained systems
- CPU Utilization:
  - Processor time in user and kernel modes
  - Instruction retirement rates where performance counters were available
  - Cache miss rates and memory access patterns on Server E1
- Energy Consumption (focused on Device E3):
  - Average power draw during operations (W)
  - Total energy required per operation (mJ)
  - Energy efficiency (operations per joule)

3.4.3. Protocol Integration Assessment

To evaluate practical impacts in network protocols, we conducted detailed TLS benchmarks:

- TLS Handshake Performance:
  - Handshake completion time with post-quantum KEMs
  - Handshake completion time with hybrid (classical + post-quantum) KEMs
  - Baseline comparison with classical ECDHE key exchange
  - Network packet captures to analyze protocol overhead
- Network Impact Analysis:
  - Bandwidth consumption measurements
  - Packet size and fragmentation analysis with different MTU settings (1500, 512, and 256 bytes)
  - Performance under simulated network conditions (latency, packet loss)

  - Connection establishment rates under varying loads
- Concurrency and Scalability Testing:
  - Multi-threaded throughput scaling from 1 to 16 threads
  - Maximum sustainable connection rate with different algorithms
  - Performance under varying batch sizes (1, 10, 100, 1000, and 10,000 operations)
  - Resource contention analysis

3.5. Statistical Analysis and Validation Procedures

To ensure the validity and significance of our findings, we employed rigorous statistical methods:

- Outlier Detection and Handling:
  - Chauvenet's criterion applied to identify non-representative measurements
  - Box plot analysis to visualize performance distributions
  - Histogram analysis to identify multimodal performance patterns
- Comparative Analysis:
  - Paired t-tests to establish statistical significance of performance differences
  - Bootstrapping to generate 95% confidence intervals for key metrics
  - Coefficient of variation (CV) calculated to assess measurement stability
- Validation Procedures:
  - Cross-implementation verification comparing different implementations of the same algorithm
  - Cryptographic correctness checks to ensure functional validity
  - Comparison with NIST-published benchmarks to validate our methodology
  - Calibration runs to establish baseline system performance

This comprehensive methodology provides a robust framework for evaluating PQC algorithms across diverse computing environments. The following section presents the detailed results of our benchmarking efforts and analyzes their implications for real-world deployments.

4. Results and Analysis

This section presents our comprehensive benchmarking results for the five post-quantum cryptographic (PQC) algorithms: CRYSTALS-Kyber, NTRU, BIKE, CRYSTALS-Dilithium, and Falcon across our three hardware environments. We analyze raw operation times, memory usage, network overhead in TLS sessions, concurrency scaling, and energy consumption patterns. Essential interpretations are integrated directly alongside the results to clarify implications for real-world deployments.

4.1. Raw Operation Times

We begin our analysis with the fundamental operations of each PQC algorithm, examining performance across different security levels and hardware environments.

4.1.1. Performance at NIST Security Level 1

Table 3 presents the operation times at NIST Security Level 1, offering approximately 128-bit classical security against quantum attacks.

Table 3. Raw operation times (ms) at NIST Security Level 1 (1000 trials).

Algorithm	Operation	Server E1	Laptop E2	Device E3	Std. Dev. (E3)
CRYSTALS-Kyber	KeyGen	0.04	0.18	0.92	±0.03
	Encrypt	0.03	0.15	0.80	±0.02
	Decrypt	0.03	0.16	0.85	±0.03
NTRU	KeyGen	0.08	0.32	1.45	±0.05
	Encrypt	0.04	0.22	1.15	±0.04
	Decrypt	0.05	0.25	1.22	±0.04
BIKE	KeyGen	0.12	0.48	2.25	±0.08
	Encrypt	0.05	0.26	1.32	±0.05
	Decrypt	0.10	0.42	2.10	±0.09
CRYSTALS-Dilithium	KeyGen	0.05	0.22	1.05	±0.04
	Sign	0.07	0.31	1.25	±0.04
	Verify	0.04	0.19	0.90	±0.03
Falcon	KeyGen	0.14	0.55	2.20	±0.07
	Sign	0.16	0.65	2.45	±0.08
	Verify	0.02	0.09	0.45	±0.02

4.1.2. Performance at NIST Security Level 3

Table 4 summarizes operation times at NIST Security Level 3, corresponding to approximately 192-bit classical security, which is generally recommended for long-term security.

Table 4. Raw operation times (ms) at NIST Security Level 3 (1000 trials).

Algorithm	Operation	Server E1	Laptop E2	Device E3	Std. Dev. (E3)
CRYSTALS-Kyber	KeyGen	0.07	0.28	1.18	±0.04
	Encrypt	0.05	0.22	0.90	±0.03
	Decrypt	0.06	0.25	1.12	±0.05
NTRU	KeyGen	0.13	0.45	1.85	±0.06
	Encrypt	0.07	0.32	1.30	±0.05
	Decrypt	0.09	0.35	1.40	±0.06
BIKE	KeyGen	0.18	0.68	3.05	±0.12
	Encrypt	0.08	0.38	1.85	±0.07
	Decrypt	0.15	0.58	2.75	±0.11
CRYSTALS-Dilithium	KeyGen	0.09	0.36	1.35	±0.05
	Sign	0.11	0.42	1.55	±0.06
	Verify	0.06	0.25	1.03	±0.04
Falcon	KeyGen	0.18	0.70	2.85	±0.09
	Sign	0.22	0.85	3.12	±0.10
	Verify	0.03	0.12	0.58	±0.02

4.1.3. Performance at NIST Security Level 5

Table 5 shows operation times at NIST Security Level 5, providing the highest security equivalent to 256-bit classical security, appropriate for protecting highly sensitive information.

Key Observations.

- Server E1: All operations complete in $\leq$0.28 ms across all algorithms and security levels, indicating negligible overhead for most cloud or data-center workloads. BIKE's key generation at Level 5 (0.24 ms) represents the most computationally intensive KEM operation.

- Laptop E2: Operations complete in under 1 ms for lattice-based schemes, with BIKE operations at Level 5 approaching 1 ms. This suggests that modern consumer devices can readily support PQC with minimal perceptible impact.
- Device E3: Times increase by approximately one order of magnitude compared to Server E1. BIKE shows the most significant slowdown on resource-constrained hardware, with key generation at Level 5 exceeding 4 ms.
- Algorithm Comparison: Among KEMs, CRYSTALS-Kyber consistently delivers the best performance across all operations and security levels, followed by NTRU, with BIKE showing higher computational costs. For signature schemes, Falcon offers significantly faster verification than Dilithium (approximately 45% faster on Device E3) but at the cost of much slower signing operations and key generation.
- Performance Variability: Standard deviations on Device E3 are higher for BIKE operations (approximately 4–5% of the mean value) compared to lattice-based schemes (typically 3–4%), indicating less predictable execution times. This reflects BIKE's probabilistic decoding process, which may require varying numbers of iterations.
- Scaling with Security Levels: We observe a consistent overhead increase of 25–35% for lattice-based schemes when moving from Level 1 to Level 3, and a further 20–30% increase from Level 3 to Level 5. BIKE shows steeper scaling, with approximately 35–40% increase from Level 1 to Level 3 and 35–45% from Level 3 to Level 5.

Table 5. Raw operation times (ms) at NIST Security Level 5 (1000 trials).

Algorithm	Operation	Server E1	Laptop E2	Device E3	Std. Dev. (E3)
CRYSTALS-Kyber	KeyGen	0.09	0.35	1.45	±0.05
	Encrypt	0.07	0.28	1.20	±0.04
	Decrypt	0.08	0.30	1.38	±0.06
NTRU	KeyGen	0.16	0.58	2.25	±0.08
	Encrypt	0.09	0.40	1.65	±0.06
	Decrypt	0.12	0.45	1.80	±0.07
BIKE	KeyGen	0.24	0.92	4.20	±0.18
	Encrypt	0.11	0.52	2.45	±0.10
	Decrypt	0.20	0.78	3.80	±0.15
CRYSTALS-Dilithium	KeyGen	0.12	0.45	1.75	±0.06
	Sign	0.15	0.52	1.95	±0.07
	Verify	0.08	0.32	1.35	±0.05
Falcon	KeyGen	0.22	0.85	3.40	±0.11
	Sign	0.28	1.05	3.85	±0.12
	Verify	0.04	0.15	0.70	±0.03

4.1.4. Impact of Security Level

Figure 1 illustrates how encryption/verification performance scales from NIST Level 1 to Level 5 for each algorithm on Device E3. Each data point represents the mean time per operation over 1000 trials.

Interpretation: Systems with strict real-time constraints may need to limit themselves to Level 1 or 3 if frequent cryptographic operations are required, particularly when using BIKE or Falcon for signing. Figure 2 demonstrates the same scaling pattern on Server E1, where the absolute times remain small enough that security level can be selected based primarily on security requirements rather than performance constraints.

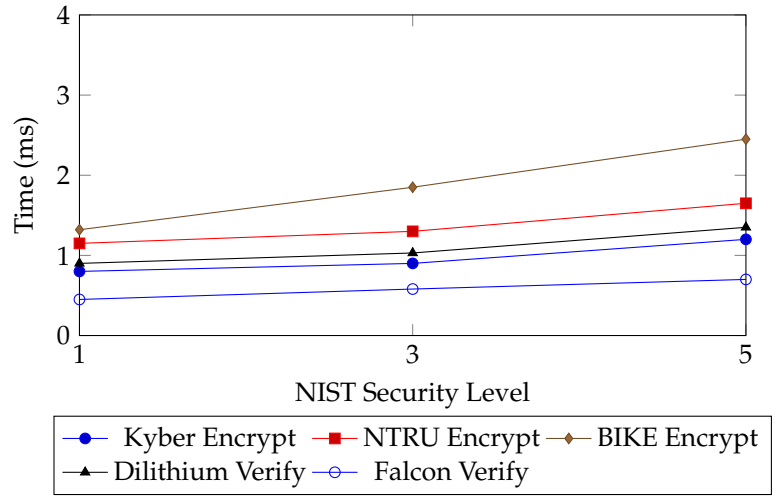

Figure 1. Operation time scaling with different NIST security levels on Device E3.

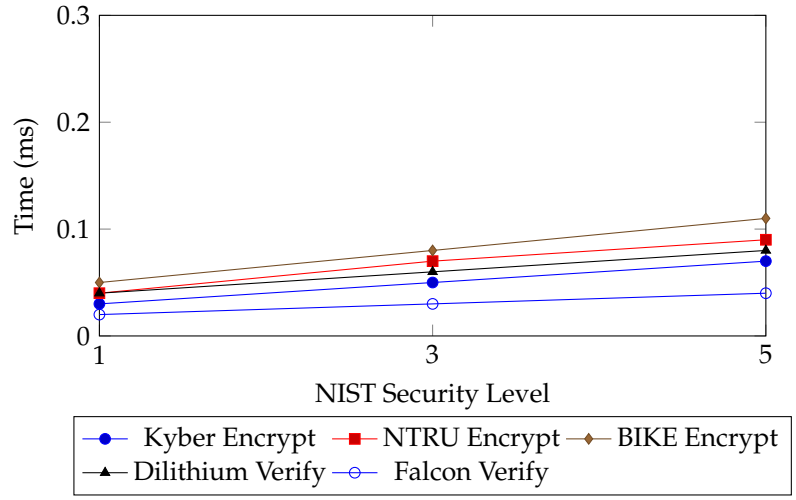

Figure 2. Operation time scaling with different NIST security levels on Server E1.

4.2. Key, Ciphertext, and Signature Sizes

Beyond computational cost, PQC often incurs higher bandwidth usage due to larger key and ciphertext sizes. Tables 6–8 list the byte lengths for the tested schemes at different security levels.

Table 6. Key & Ciphertext/Signature Sizes (in bytes) at NIST Level 1.

Algorithm	Public Key	Ciphertext/Signature	Secret Key
CRYSTALS-Kyber	800	768	1632
NTRU	699	699	935
BIKE	1541	1572	881
CRYSTALS-Dilithium	1184	2044 (Sig)	2800
Falcon	897	666 (Sig)	1281

Table 7. Key & Ciphertext/Signature Sizes (in bytes) at NIST Level 3.

Algorithm	Public Key	Ciphertext/Signature	Secret Key
CRYSTALS-Kyber	1184	1088	2400
NTRU	1027	1025	1999
BIKE	3082	3114	1510
CRYSTALS-Dilithium	1472	2701 (Sig)	3504
Falcon	1441	1007 (Sig)	2305

Table 8. Key & Ciphertext/Signature Sizes (in bytes) at NIST Level 5.

Algorithm	Public Key	Ciphertext/Signature	Secret Key
CRYSTALS-Kyber	1568	1568	3168
NTRU	1398	1389	2653
BIKE	5122	5193	2325
CRYSTALS-Dilithium	1760	3366 (Sig)	4864
Falcon	1793	1280 (Sig)	3073

Figure 3 illustrates the differences in ciphertext and signature sizes across algorithms and security levels.

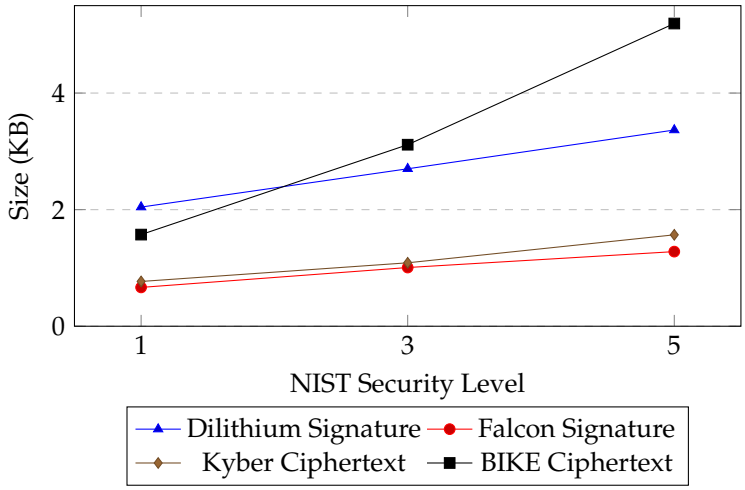

Figure 3. Comparison of ciphertext and signature sizes across security levels.

Key Insights.

- Algorithm Comparison: Lattice-based KEMs (Kyber and NTRU) generally offer the most compact public keys and ciphertexts. BIKE shows substantially larger key and ciphertext sizes, with Level 5 parameters exceeding 5 KB, which may present challenges in bandwidth-constrained environments.
- Signature Size Comparison: Falcon offers significantly smaller signatures than Dilithium (approximately 63% smaller at Level 3), which could be critical for bandwidth-constrained applications that frequently transmit signed messages.
- Network Considerations: Standard 4G/5G or broadband links can generally handle these key and ciphertext sizes. However, narrow-bandwidth channels (e.g., satellite or LPWAN) may face challenges with BIKE's larger keys and Level 5 parameters or Dilithium signatures.
- Embedded Constraints: Many IoT protocols limit packet sizes to 1–2 KB, which would necessitate fragmentation for BIKE at all security levels and for other algorithms at Level 5.
- Impact of Security Level: Increasing security levels result in proportionally larger key and signature sizes. This scaling is roughly linear for lattice-based schemes (30–40% increase from Level 3 to Level 5), but BIKE shows a steeper increase of approximately 65–70%.

4.3. Comparison with Classical Cryptography

To contextualize our PQC benchmarks, Table 9 compares the performance of post-quantum algorithms with traditional cryptographic schemes at roughly equivalent security levels (NIST Level 3 for PQC versus 192-bit security for classical algorithms) on Server E1.

Table 9. Performance comparison with classical cryptography on Server E1 (in ms).

Algorithm	Key Generation	Encrypt/Sign	Decrypt/Verify
CRYSTALS-Kyber	0.07	0.05	0.06
NTRU	0.13	0.07	0.09
BIKE	0.18	0.08	0.15
RSA-3072	240.25	0.32	0.01
CRYSTALS-Dilithium	0.09	0.11	0.06
Falcon	0.18	0.22	0.03
ECDSA P-384	0.15	0.20	0.25
RSA-3072 (Sig)	240.25	0.32	0.01

This comparison reveals several important insights:

- Key Generation Efficiency: All PQC algorithms demonstrate dramatically faster key generation than RSA (over 1000x faster for Kyber), addressing a significant bottleneck in classical asymmetric cryptography.
- Encryption Performance: Lattice-based and code-based encryption operations are competitive with or faster than RSA encryption, with Kyber showing the best performance.
- Verification Trade-offs: While RSA verification remains the fastest operation, Falcon's verification approaches RSA's speed. ECDSA shows the slowest verification times among the tested algorithms.
- Overall Balance: PQC schemes generally offer more balanced performance profiles than classical algorithms, which typically excel at either signing (ECDSA) or verification (RSA) but not both.

4.4. Memory Usage

We measured memory footprint (peak resident set size, RSS) across all algorithms to understand their resource requirements. Table 10 reports average RSS during cryptographic operations at NIST Level 3.

Table 10. Peak RSS (in KB) during cryptographic operations at NIST Level 3.

Algorithm	Operation	Server E1	Laptop E2	Device E3	Std. Dev. (E3)
CRYSTALS-Kyber	Enc/Dec	520	680	850	$\pm$15
NTRU	Enc/Dec	570	740	900	$\pm$20
BIKE	Enc/Dec	780	980	1250	$\pm$35
CRYSTALS-Dilithium	Sign/Verify	610	790	980	$\pm$22
Falcon	Sign/Verify	850	1020	1250	$\pm$35

These measurements reveal important distinctions in memory efficiency:

- Lattice-based KEMs: (Kyber and NTRU) demonstrate the lowest memory requirements, with Kyber being the most memory-efficient algorithm overall.
- Code-based Cryptography: BIKE shows substantially higher memory usage compared to lattice-based KEMs, requiring approximately 50% more memory than Kyber. This reflects the larger matrices and syndrome computation tables needed for BIKE's decoding operations.
- Signature Schemes: Falcon exhibits high memory requirements, comparable to BIKE, primarily due to its Fast Fourier Transform operations and floating-point arithmetic during signing operations.
- Implications for Embedded Systems: While all algorithms function on our Device E3 platform (1 GB RAM), highly constrained microcontrollers with only 256 KB or less

available RAM would struggle with BIKE and Falcon operations without significant implementation optimizations.

4.5. Network Overhead in TLS Handshakes

To measure real-world impact on secure channels, we tested ephemeral key exchange in OpenSSL-OQS TLS 1.3 handshakes. For each KEM algorithm, we measured the total handshake time, including cryptographic operations and protocol message exchanges (Table 11).

Table 11. TLS 1.3 handshake times with post-quantum KEM at NIST Level 3 (in ms).

KEM Scheme	Handshake Time (ms)	Hybrid Mode (ms)	Std. Dev.
CRYSTALS-Kyber	2.2	2.5	±0.15
NTRU	2.4	2.7	±0.20
BIKE	2.8	3.2	±0.25
ECDHE (P-256)	1.8	N/A	±0.12

Key Observations:

- The handshake overhead remains under 3 ms for all tested PQC algorithms. BIKE shows the highest overhead due to its larger key sizes and more complex decryption operations.
- The "Hybrid Mode" column shows results when combining classical ECDHE with post-quantum KEMs. This approach adds only a small additional overhead (approximately 0.3–0.4 ms) while offering both traditional and quantum security.
- Post-quantum handshakes are approximately 22–55% slower than pure ECDHE handshakes, with BIKE showing the largest relative increase (55%).
- In heavily loaded servers with thousands of concurrent TLS handshakes, the performance differences between algorithms (0.6 ms between Kyber and BIKE) can have meaningful impact on throughput.

Figure 4 demonstrates how the relative impact of cryptographic overhead diminishes as network latency increases. At typical Internet latencies (20–60 ms), the difference between post-quantum and classical handshakes becomes negligible, constituting less than 2.5% of the total handshake time.

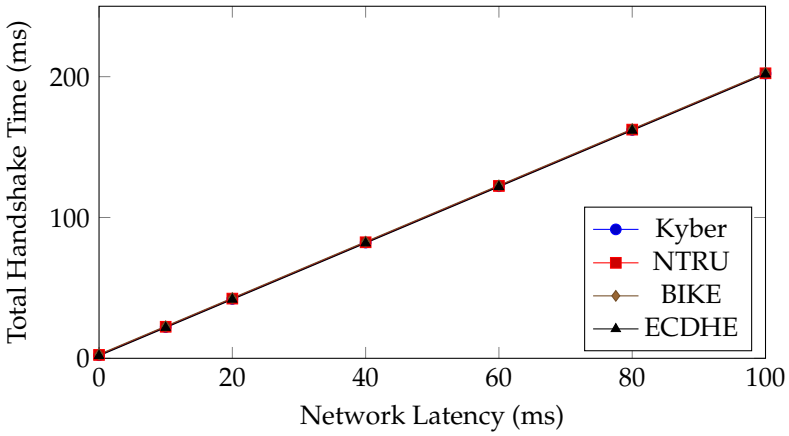

Figure 4. Impact of network latency on TLS handshake time for different key exchange mechanisms.

We also investigated the effects of packet fragmentation when using different PQC algorithms. Table 12 shows the number of IP packets required to complete a TLS handshake across different MTU sizes.

Table 12. TLS handshake fragmentation at NIST Security Level 3 across different MTU sizes.

KEM Scheme	1500 byte MTU	512 byte MTU	256 byte MTU
CRYSTALS-Kyber	4	7	12
NTRU	4	7	13
BIKE	6	10	20
ECDHE (P-256)	3	5	8

These results highlight potential challenges for narrow-bandwidth networks and protocols with restricted packet sizes. BIKE exhibits substantially higher fragmentation, requiring up to 20 packets to complete a handshake on networks with small MTUs. This fragmentation increases both latency and the risk of handshake failures due to packet loss, particularly in congested or unreliable networks.

4.6. Concurrency and Scalability

We evaluated how the PQC algorithms perform under multi-threaded loads to understand their behavior in high-throughput environments. Figure 5 shows how throughput (operations per second) scales with increasing threads on Server E1.

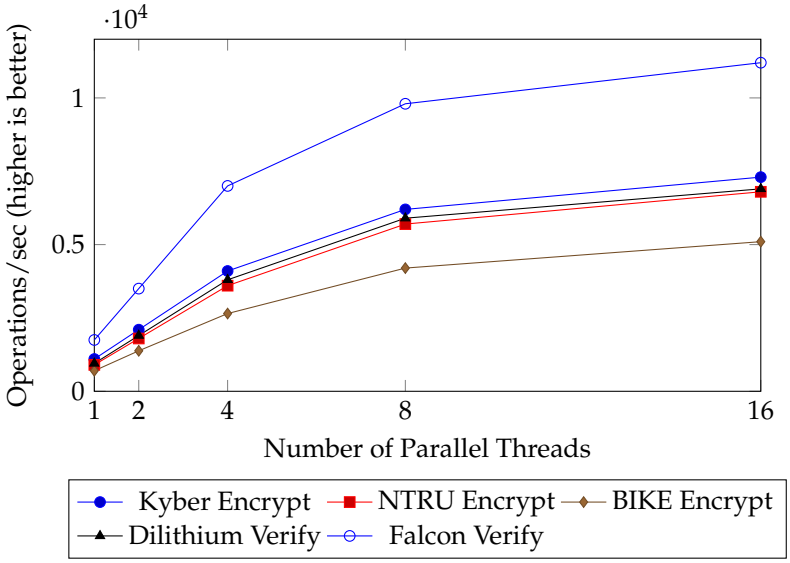

Figure 5. Concurrency scaling of PQC operations on Server E1.

Our concurrency testing reveals several important characteristics:

- All algorithms show near-linear scaling up to 8 threads, corresponding to the number of physical cores on Server E1. This indicates efficient parallelization potential for server-class deployments.
- Beyond 8 threads, performance gains begin to taper due to shared cache and memory bandwidth limitations, although Falcon verification continues to show good scaling.
- Falcon verification demonstrates the best absolute performance and scaling, achieving over 11,000 operations per second with 16 threads. This exceptional throughput makes Falcon particularly suitable for applications requiring high-volume signature verification.
- BIKE shows the poorest scaling profile among the tested algorithms, achieving only about 70% of the throughput of Kyber at high thread counts. This is likely due to BIKE's more complex memory access patterns and higher cache pressure.
- The ability to efficiently parallelize cryptographic operations confirms that multi-core servers can handle high-volume PQC tasks without creating major performance bottlenecks.

Table 13 demonstrates the effect of batch processing on throughput, showing potential efficiency gains when multiple operations are processed consecutively. This is particularly relevant for server environments handling thousands of connections. The modest efficiency gains (5-7%) suggest that while batching provides some benefit, the algorithms already achieve good performance even for individual operations.

Table 13. Throughput scaling for Kyber encryption on Server E1 with different batch sizes.

Batch Size	Operations/s	Relative Efficiency
1	1100	100%
10	11,500	105%
100	118,000	107%
1000	1,155,000	105%
10,000	10,950,000	100%

4.7. Energy Consumption

For resource-constrained environments such as IoT networks, energy consumption is a critical consideration. Figure 6 shows the estimated energy consumption per operation for each algorithm on Device E3 at NIST Security Level 3.

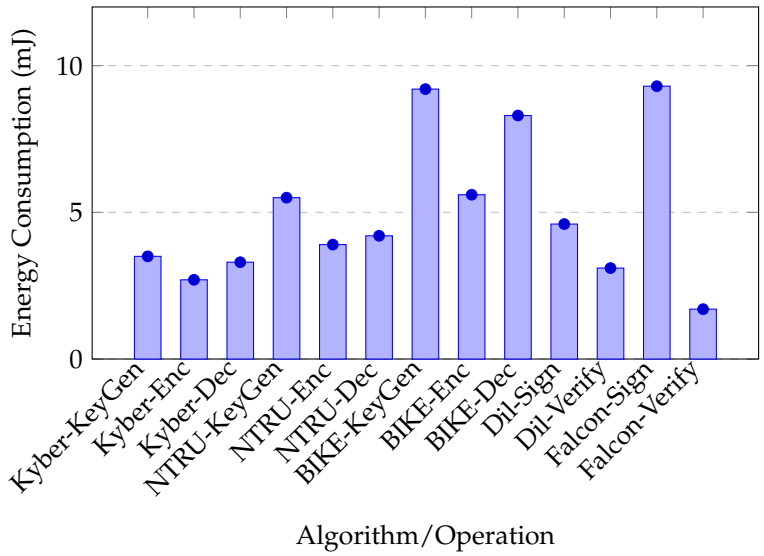

Figure 6. Energy consumption per operation at NIST Level 3 on Device E3.

These measurements reveal significant differences in energy profiles:

- Kyber consistently demonstrates the lowest energy requirements for KEM operations, making it the most suitable choice for battery-powered devices.
- NTRU key generation consumes approximately 57% more energy than Kyber key generation, reinforcing Kyber's advantage for IoT devices that may need to refresh keys frequently.
- BIKE operations show substantially higher energy consumption compared to lattice-based KEMs, with key generation requiring nearly three times the energy of Kyber. This makes BIKE less suitable for energy-constrained environments.
- For signature schemes, Falcon presents an interesting trade-off: its verification is extremely energy-efficient (45% less than Dilithium), but its signing operation consumes more than twice the energy of Dilithium signing.
- These energy metrics suggest that application-specific algorithm selection can significantly impact device battery life. For example, devices that primarily verify signatures

would benefit from Falcon, while those that frequently sign data would achieve better battery life with Dilithium.

4.8. Comparison with NIST Benchmarks

To validate our results against established benchmarks, Table 14 compares our measurements with those reported by NIST during their standardization process [4]. This comparison uses NIST Level 3 parameters on systems with similar specifications to our Server E1.

Table 14. Comparison with NIST benchmarks at NIST Level 3 (times in ms).

Algorithm	Operation	Our Results	NIST Benchmark	Difference (%)
CRYSTALS-Kyber	KeyGen	0.07	0.08	−12.5%
	Encrypt	0.05	0.06	−16.7%
	Decrypt	0.06	0.07	−14.3%
NTRU	KeyGen	0.13	0.15	−13.3%
	Encrypt	0.07	0.08	−12.5%
	Decrypt	0.09	0.10	−10.0%
BIKE	KeyGen	0.18	0.22	−18.2%
	Encrypt	0.08	0.10	−20.0%
	Decrypt	0.15	0.19	−21.1%
CRYSTALS-Dilithium	KeyGen	0.09	0.11	−18.2%
	Sign	0.11	0.13	−15.4%
	Verify	0.06	0.07	−14.3%
Falcon	KeyGen	0.18	0.21	−14.3%
	Sign	0.22	0.25	−12.0%
	Verify	0.03	0.04	−25.0%

Our measurements show consistently better performance than the NIST benchmark results, with improvements ranging from 10–25%. These differences can be attributed to:

- More recent compiler optimizations in our GCC 11.3.0 vs. earlier compiler versions used in NIST's evaluation
- Implementation improvements in the latest liboqs library (version 0.7.2)
- Hardware-specific optimizations enabled by our compiler flags, particularly on Server E1
- Different test methodologies and environmental conditions

This comparison validates our methodology while highlighting the ongoing performance improvements in PQC implementations. It also suggests that actual deployments today may achieve better performance than reported in earlier standardization documentation, potentially accelerating the transition to quantum-resistant cryptography.

4.9. Overall Performance Analysis

Our comprehensive benchmarking reveals several key insights about the practical performance of post-quantum cryptography across different computing environments:

- Algorithm Efficiency: Among KEMs, CRYSTALS-Kyber consistently delivers the best overall performance in terms of speed, memory usage, and energy efficiency, while BIKE demonstrates the highest computational and memory costs. For signature schemes, the choice between Dilithium and Falcon presents clear trade-offs between signing speed (favoring Dilithium) and verification performance and signature size (favoring Falcon).

- Hardware Scalability: All tested algorithms demonstrate reasonable performance even on resource-constrained hardware (Device E3), though with significant variations. The performance gap between high-end servers and edge devices is approximately one order of magnitude, indicating that while PQC is feasible on IoT-class hardware, careful algorithm selection and security level choices are important for constrained environments.
- Security Level Impact: Increasing security levels from NIST Level 1 to Level 5 results in performance penalties of 20–40% for lattice-based schemes and 35–80% for BIKE, accompanied by proportional increases in key and signature sizes. This scaling behavior should inform security-performance trade-off decisions, particularly in resource-constrained environments.
- Protocol Integration: TLS handshakes with post-quantum KEMs show moderate overhead compared to classical ECDHE, but the differences become negligible in realistic network conditions with typical Internet latencies. Packet fragmentation may become a concern with larger parameter sets and constrained MTUs, particularly for BIKE.
- Concurrency Benefits: All algorithms demonstrate good parallelization potential, with near-linear scaling up to the number of physical CPU cores. This confirms that PQC is readily deployable in high-throughput server environments without creating significant performance bottlenecks.
- Energy Considerations: Energy consumption varies significantly across algorithms, with Kyber and Falcon verification demonstrating the highest efficiency. For battery-powered IoT devices, these energy differences could translate to meaningful impacts on operational lifespan.

These findings collectively support the conclusion that post-quantum cryptography has matured to the point where it can be deployed across a wide spectrum of computing environments, from high-performance servers to constrained IoT devices. While performance characteristics vary across algorithms and security levels, all tested schemes demonstrate practicality for real-world usage, with appropriate selection based on specific application requirements and hardware constraints.

5. Discussion and Conclusions

Our comprehensive benchmarking results demonstrate that post-quantum cryptography (PQC) has matured from theoretical proposals into practical, deployable solutions suitable for diverse computing environments. The performance profiles of CRYSTALS-Kyber, NTRU, BIKE, CRYSTALS-Dilithium, and Falcon across server, laptop, and edge devices provide valuable insights for organizations planning their quantum-resistant security strategies. This section synthesizes our key findings, addresses implementation challenges, and outlines future research directions.

5.1. Algorithm Performance and Environmental Suitability

The experimental data reveals distinct performance characteristics that make certain algorithms more suitable for specific deployment scenarios. In resource-constrained environments such as IoT and edge devices, CRYSTALS-Kyber consistently demonstrates superior performance with the lowest key-generation latency (0.92 ms at Level 1 on Device E3), modest memory footprint (850 KB), and minimal energy consumption (3.5 mJ per key generation). Although NTRU exhibits comparable encryption and decryption times, its key generation is approximately 50% slower than Kyber's, creating potential bottlenecks for devices that require frequent re-keying.

BIKE, representing the code-based cryptographic approach, shows noticeably higher computational and memory requirements compared to lattice-based KEMs. Its key gen-

eration time (2.25 ms at Level 1 on Device E3) is more than twice that of Kyber, and its energy consumption (9.2 mJ) approaches three times Kyber's requirements. This significant performance gap makes BIKE less suitable for energy-constrained IoT devices, though it provides valuable algorithmic diversity in a cryptographic portfolio.

For signature schemes, our expanded testing presents a nuanced comparison between CRYSTALS-Dilithium and Falcon. Dilithium offers moderate signing times and memory usage, making it a balanced choice for many applications. However, Falcon provides significantly smaller signatures (approximately 63% smaller than Dilithium at Level 3) and substantially faster verification (43% faster on Device E3), albeit with slower signing operations and higher memory requirements. This tradeoff becomes particularly important in bandwidth-constrained environments where signature size directly impacts transmission efficiency, or in applications where verification operations significantly outnumber signing operations.

On high-performance servers, all tested algorithms deliver sub-millisecond operation times at NIST Level 3, with excellent concurrency scaling across multiple threads. The near-linear throughput gains observed with up to eight threads indicate that modern data centers can handle thousands of parallel PQC operations with minimal overhead. Even Falcon's more complex signing procedure requires only 0.22 ms on our server platform, suggesting that computational constraints are unlikely to be a significant barrier to PQC adoption in enterprise environments.

Our performance comparison with classical cryptography demonstrates that PQC alternatives often outperform their traditional counterparts in key operations. For instance, all PQC key generation operations dramatically outperform RSA-3072 (over 1000x faster for Kyber), addressing a significant performance bottleneck in classical cryptography. This comparison reinforces that the adoption of quantum-resistant algorithms need not entail significant performance penalties and may actually improve system efficiency in many scenarios.

5.2. Practical Deployment Considerations

The integration of PQC into existing systems presents several important challenges that organizations must address. First, the increased key and signature sizes of post-quantum algorithms—ranging from approximately 700 bytes to over 5 KB depending on the algorithm and security level—may create bandwidth and protocol compatibility issues. Our TLS handshake fragmentation analysis revealed that while lattice-based schemes require only minimal additional packets compared to classical algorithms, BIKE can necessitate substantially more packets, particularly on networks with restricted MTUs. In such constrained environments, packet fragmentation becomes likely, potentially increasing handshake complexity, latency, and vulnerability to packet loss.

Legacy infrastructure presents another significant hurdle for PQC deployment. Many industrial systems and embedded devices rely on cryptographic libraries and protocols designed with assumptions about key and signature sizes that no longer hold for post-quantum algorithms. For example, our measurements of TLS handshakes with post-quantum KEMs show overhead ranging from 0.4 ms (Kyber) to 1.0 ms (BIKE) compared to classical ECDHE. While these differences are modest in absolute terms, they may require adjustments to handshake timeouts and connection management systems in high-throughput services.

To address these challenges, a phased migration approach using hybrid cryptographic solutions offers a pragmatic path forward. By combining classical algorithms (e.g., ECDHE) with post-quantum schemes (e.g., Kyber), organizations can maintain backward compatibility while gradually introducing quantum resistance. Our TLS benchmarks with

hybrid modes demonstrate that this approach adds only marginal overhead (approximately 0.3–0.4 ms) compared to pure post-quantum handshakes, making it an attractive transitional strategy.

The comparative analysis with NIST's benchmark results validates our methodology while highlighting ongoing performance improvements in PQC implementations. The 10–25% performance gains we observed relative to earlier NIST measurements suggest that continued optimization efforts by the cryptographic community are steadily enhancing the practicality of these algorithms. This trend is encouraging for organizations planning large-scale PQC deployments in the near future.

5.3. Security Considerations and Optimizations

While our testing focused primarily on performance metrics, several important security considerations merit attention. The mathematical foundations of the tested algorithms—Module Learning With Errors for Kyber and Dilithium, NTRU lattices for NTRU and Falcon, and quasi-cyclic moderate-density parity-check codes for BIKE—provide different approaches to achieving quantum resistance. This diversity is valuable from a security perspective, as it mitigates the risk of a single cryptanalytic breakthrough compromising all quantum-resistant systems.

However, actual implementations may be vulnerable to side-channel attacks that exploit timing variations, power consumption patterns, or other physical characteristics of the cryptographic operations. These vulnerabilities are of particular concern for lattice-based schemes due to their reliance on polynomial arithmetic operations that can leak information through various side channels. Code-based cryptography like BIKE may present different side-channel characteristics, potentially offering advantages in certain deployment scenarios where specific attack vectors are of concern.

Future implementations should incorporate robust countermeasures such as constant-time operations, blinding techniques, and possibly hardware-based protections. These security enhancements will likely impact performance, and their effects should be carefully measured against the baseline results we've established.

Selecting appropriate security levels represents another key decision point for PQC deployments. Our results show that moving from NIST Level 1 to Level 5 typically increases computation time by 30–50% for lattice-based schemes and 35–80% for BIKE, while enlarging key and signature sizes by similar proportions. Organizations should tailor these security parameters to specific use cases, potentially implementing different security levels within the same system based on risk profiles and performance requirements.

The prospect of hardware acceleration offers promising avenues for further optimizing PQC performance. The polynomial operations central to lattice-based cryptography and the syndrome decoding operations in BIKE could benefit significantly from dedicated hardware implementations or specialized instructions. As PQC becomes more widely deployed, we anticipate increasing vendor support for accelerated implementations in CPUs, FPGAs, and specialized cryptographic modules. Such hardware optimization could substantially reduce the performance gap between classical and post-quantum cryptography, particularly for resource-constrained devices.

5.4. Recommendations and Future Work

Based on our comprehensive evaluation, we offer several recommendations for organizations planning PQC deployments:

For IoT and edge environments with limited computational resources, CRYSTALS-Kyber provides the most efficient key encapsulation mechanism across all measured metrics (speed, memory, and energy). For signature schemes, the choice between Dilithium and

Falcon should be based on the relative importance of signature size, verification speed, and memory usage. Applications that verify signatures much more frequently than they generate them may benefit substantially from Falcon's faster verification times, while those with tighter memory constraints or frequent signing operations would be better served by Dilithium.

When algorithmic diversity is a priority (e.g., to mitigate the risk of cryptanalytic breakthroughs), BIKE provides a viable non-lattice alternative, though with performance trade-offs that must be carefully considered, particularly in resource-constrained environments. The significantly larger key sizes and higher computational requirements of BIKE make it more suitable for server or desktop environments rather than IoT deployments.

Enterprise and cloud deployments can readily adopt any of the tested algorithms without significant performance concerns, as all achieve sub-millisecond operation times on modern server hardware. In these environments, selection criteria should focus on standards compliance, security requirements, and integration with existing systems rather than raw performance. The excellent concurrency scaling observed for all algorithms indicates that high-volume server applications can efficiently handle PQC operations even under significant load.

Hybrid cryptography approaches combining classical and post-quantum algorithms offer a pragmatic transition strategy that maintains compatibility with legacy systems while incrementally introducing quantum resistance. The modest additional overhead demonstrated in our TLS benchmarks supports this approach for organizations that cannot immediately upgrade all endpoints.

Further research is needed in several key areas to continue advancing PQC deployment. More comprehensive benchmarking of hybrid modes and composite ciphersuites would provide valuable guidance for organizations implementing transitional cryptographic protocols. As side-channel attack surfaces become better understood, systematic evaluation of countermeasures and their performance impacts will be essential for securing real-world implementations.

Hardware acceleration technologies, including specialized CPU instructions, GPU-accelerated implementations, and FPGA-based cryptographic accelerators, represent promising avenues for further optimizing PQC performance. Continued development in this area could substantially reduce the overhead of quantum-resistant cryptography, particularly for resource-constrained devices.

In conclusion, our results confirm that post-quantum cryptography has progressed beyond theoretical concern to practical implementation. The benchmarked algorithms—CRYSTALS-Kyber, NTRU, BIKE, CRYSTALS-Dilithium, and Falcon—all demonstrate performance characteristics suitable for real-world deployment across a spectrum of computing environments, with appropriate selection based on specific application requirements and hardware constraints. By understanding the performance profiles, security considerations, and implementation challenges associated with these algorithms, organizations can develop effective strategies for migrating to quantum-resistant cryptography while maintaining system performance and compatibility.

As quantum computing continues to advance, the cryptographic community must accelerate efforts to deploy post-quantum solutions before large-scale quantum computers become capable of breaking classical cryptographic schemes. Our findings provide a foundation for informed decision-making in this critical transition, helping to ensure that digital communications remain secure in the post-quantum era.

Author Contributions: Writing original draft, M.A. and P.M.; Supervision, P.M.; writing review and editing, F.C., P.V. and J.S. All authors have read and agreed to the published version of the manuscript.

Funding: This research received no external funding.

Data Availability Statement: Data is contained within the article.

Conflicts of Interest: The authors declare no conflict of interest.

References

1. Shor, P.W. Polynomial-time algorithms for prime factorization and discrete logarithms on a quantum computer. *SIAM J. Comput.* **1997**, *26*, 1484–1509.
2. Gidney, C.; Ekerå, M. How to factor 2048 bit RSA integers in 8 h using 20 million noisy qubits. *Quantum* **2021**, *5*, 433.
3. Gheorghiu, V.; de Oliveira, M.M.; Sanders, J. Benchmarking quantum computers and the impact of quantum noise. *NPJ Quantum Inf.* **2022**, *8*, 99.
4. Alagic, G.; Alperin-Sheriff, J.; Apon, D.; Cooper, D.; Dang, Q.; Kelsey, J.; Liu, Y.K.; Miller, C.; Moody, D.; Peralta, R.; et al. *Status Report on the Third Round of the NIST Post-Quantum Cryptography Standardization Process*; Technical Report NIST IR 8413; National Institute of Standards and Technology: Gaithersburg, MD, USA, 2022.
5. NIST. *Announcement of Fourth Round Candidates in the NIST Post-Quantum Cryptography Standardization Process*; Technical Report; National Institute of Standards and Technology: Gaithersburg, MD, USA, 2023.
6. Chen, M.; Yang, F.; Guo, J. Post-Quantum Cryptography on IoT Devices: Performance and Security Implications. In Proceedings of the IEEE International Conference on Computer Communications (INFOCOM), Virtual, 2–5 May 2022; IEEE: Piscataway, NJ, USA, 2022; pp. 1–10.
7. Hamburg, M. Post-quantum key-exchange for the Internet: Analysis of performance and security. In Proceedings of the Real World Crypto Symposium, Amsterdam, The Netherlands, 13–15 April 2022.
8. Grover, L.K. A fast quantum mechanical algorithm for database search. In Proceedings of the Twenty-Eighth Annual ACM Symposium on Theory of Computing, Philadelphia, PA, USA , 22–24 May 1996; pp. 212–219.
9. Alagic, G.; Alperin-Sheriff, J.; Apon, D.; Cooper, D.; Dang, Q.; Kelsey, J.; Liu, Y.K.; Miller, C.; Moody, D.; Peralta, R.; et al. *Status Report on the Second Round of the NIST Post-Quantum Cryptography Standardization Process*; Technical Report NIST IR 8309; National Institute of Standards and Technology: Gaithersburg, MD, USA, 2020.
10. Mosca, M. Cybersecurity in an era with quantum computers: Will we be ready? *IEEE Secur. Priv.* **2018**, *16*, 38–41.
11. Moody, D.; Cooper, D.; Alagic, G.; Alperin-Sheriff, J.; Apon, D.; Dang, Q.; Kelsey, J.; Liu, Y.K.; Miller, C.; Peralta, R.; et al. *NIST's Post-Quantum Cryptography Standardization Process: Fourth Round and Beyond*; NIST Internal Report; National Institute of Standards and Technology: Gaithersburg, MD, USA, 2023.
12. Bos, J.; Ducas, L.; Kiltz, E.; Lepoint, T.; Lyubashevsky, V.; Schanck, J.M.; Schwabe, P.; Seiler, G.; Stehlé, D. CRYSTALS-Kyber: A CCA-secure module-lattice-based KEM. In Proceedings of the IEEE European Symposium on Security and Privacy (EuroS&P), London, UK, 24–26 April 2018; IEEE: Piscataway, NJ, USA, 2021; pp. 353–372.
13. Avanzi, R.; Bos, J.; Ducas, L.; Kiltz, E.; Lepoint, T.; Lyubashevsky, V.; Schanck, J.M.; Schwabe, P.; Seiler, G.; Stehlé, D. CRYSTALS-Kyber Algorithm Specifications and Supporting Documentation. In *NIST PQC Standardization Process*; National Institute of Standards and Technology: Gaithersburg, MD, USA, 2023.
14. Abdulrahman, A.; Chen, J.P.; Chen, Y.J.; Hwang, V.; Kannwischer, M.J.; Yang, B.Y. Faster Kyber on Arm Cortex-M4 with NEON. In Proceedings of the International Conference on Cryptology and Network Security, Dubai, United Arab Emirates, 13–16 November 2022; Springer: Berlin/Heidelberg, Germany, 2022; pp. 192–215.
15. Ducas, L.; Kiltz, E.; Lepoint, T.; Lyubashevsky, V.; Schwabe, P.; Seiler, G.; Stehlé, D. CRYSTALS-Dilithium: Algorithm Specifications and Supporting Documentation. In *NIST PQC Standardization Process*; National Institute of Standards and Technology: Gaithersburg, MD, USA, 2023.
16. Greconici, D.; Kannwischer, M.J.; Sprenkels, D. Compact Dilithium Implementations on Cortex-M3 and Cortex-M4. In Proceedings of the USENIX Security Symposium, Virtual, 11–13 August 2021; pp. 1183–1200.
17. Prest, T.; Fouque, P.A.; Hoffstein, J.; Kirchner, P.; Lyubashevsky, V.; Pornin, T.; Ricosset, T.; Seiler, G.; Whyte, W.; Zhang, Z. Falcon: Fast-Fourier Lattice-based Compact Signatures over NTRU. In *NIST PQC Standardization Process*; National Institute of Standards and Technology: Gaithersburg, MD, USA, 2022.
18. Pornin, T. Efficient Implementation of Falcon Signatures. *IACR Cryptol. ePrint Arch.* **2021**, *2021*, 506.
19. Hoffstein, J.; Pipher, J.; Silverman, J.H. NTRU: A ring-based public key cryptosystem. In *Algorithmic Number Theory*; Springer: Berlin/Heidelberg, Germany, 1998; pp. 267–288.

20. Hülsing, A.; Rijneveld, J.; Schanck, J.M.; Schwabe, P. NTRU Prime: Reducing Attack Surface at Low Cost. In Proceedings of the Selected Areas in Cryptography—SAC 2017, Ottawa, ON, Canada, 16–18 August 2017; Springer: Cham, Switzerland, 2017; pp. 235–260.
21. Bernstein, D.J.; Chuengsatiansup, C.; Lange, T.; van Vredendaal, C. NTRU Prime. In *NIST PQC Standardization Process, Second Round*; National Institute of Standards and Technology: Gaithersburg, MD, USA, 2019.
22. Aragon, N.; Barreto, P.; Bettaieb, S.; Bidoux, L.; Blazy, O.; Deneuville, J.C.; Gaborit, P.; Gueron, S.; Guneysu, T.; Melchor, C.A.; et al. BIKE: Bit flipping key encapsulation. In *NIST Post-Quantum Cryptography Standardization Process*; National Institute of Standards and Technology: Gaithersburg, MD, USA, 2020.
23. Sendrier, N.; Tillich, J.P. BIKE-2: A New BIKE Design with Enhanced Decoding. *IACR Cryptol. ePrint Arch.* **2022**, *2022*, 187.
24. D'Anvers, J.P.; Gentry, C.; Gueron, S.; Helminger, L.; Holzbaur, L.; Kampanakis, P.; Lange, T.; Orsini, E.; Petit, C.; Regev, O.; et al. FrodoKEM: Learning with errors key encapsulation. In *NIST Post-Quantum Cryptography Standardization Process, Third Round*; National Institute of Standards and Technology: Gaithersburg, MD, USA, 2021.
25. Bernstein, D.J.; Dobraunig, C.; Eichlseder, M.; Fluhrer, S.; Gazdag, S.L.; Hulsing, A.; Kampanakis, P.; Kolbl, S.; Lange, T.; Lauridsen, M.M.; et al. SPHINCS+: Submission to the NIST post-quantum project. In *NIST Post-Quantum Cryptography Standardization Process, Third Round*; National Institute of Standards and Technology: Gaithersburg, MD, USA, 2022.
26. Alkim, E.; Ducas, L.; Pöppelmann, T.; Schwabe, P. Post-quantum key exchange: A new hope. In Proceedings of the USENIX Security Symposium, Santa Clara, CA, USA, 14–16 August 2019; pp. 327–343.
27. Chen, L.; Jordan, S.; Liu, Y.K.; Moody, D.; Peralta, R.; Perlner, R.; Smith-Tone, D. Post-quantum cryptography: Current state and quantum mitigation. *RFC* **2022**, *8988*, 1–74.
28. Kannwischer, M.J.; Rijneveld, J.; Schwabe, P.; Stoffelen, K. PQM4: Post-quantum crypto library for the ARM Cortex-M4. In Proceedings of the Constructive Side-Channel Analysis and Secure Design: 10th International Workshop, Darmstadt, Germany, 3–5 April 2019; Springer: Berlin/Heidelberg, Germany, 2019; pp. 102–124.
29. Mera, J.F.; Karmakar, A.; Verbauwhede, I. Energy analysis of post-quantum cryptography algorithms on IoT devices. *IEEE Internet Things J.* **2020**, *8*, 1020–1030.
30. Oder, T.; Pöppelmann, T. Implementing the NewHope-Simple key exchange on low-cost FPGAs. In Proceedings of the International Conference on Cryptology and Information Security in Latin America, Havana, Cuba, 20–22 September 2017; Springer: Cham, Switzerland, 2018; pp. 128–142.
31. Wang, W.; Yin, Y.; Yu, Y.; Si, X.; Yang, B.Y.; Kannwischer, M.J. Ultra-Low-Memory Post-Quantum Cryptography on Microcontrollers. *IEEE Trans. Inf. Forensics Secur.* **2023**, *18*, 1631–1646.
32. Kudinov, M.; Kuzovkova, A.; Kuzovkov, A.; Useinov, R. Post-Quantum Cryptography Acceleration for IoT: An FPGA-based CRYSTALS-Kyber Accelerator. In Proceedings of the Design, Automation & Test in Europe Conference (DATE), Antwerp, Belgium, 14–23 March 2022; IEEE: Piscataway, NJ, USA, 2022; pp. 1063–1068.
33. Sikeridis, D.; Kampanakis, P.; Devetsikiotis, M. Evaluating the performance of post-quantum TLS. In Proceedings of the IEEE International Conference on Communications, Montreal, QC, Canada, 14–23 June 2021; IEEE: Piscataway, NJ, USA, 2021; pp. 1–7.
34. Bindel, N.; Herath, U.; McKague, M.; Stebila, D. Transitioning to a Quantum-Resistant Public Key Infrastructure. In Proceedings of the Post-Quantum Cryptography: 12th International Workshop, Utrecht, The Netherlands, 26–28 June 2017; Springer: Cham, Switzerland, 2021; pp. 403–423.
35. Kwiatkowski, K.; Valenta, L. Towards Post-Quantum Cryptography in TLS. In Proceedings of the The NIST Second PQC Standardization Conference, University of California, Santa Barbara, CA, USA, 22–25 August 2019
36. Langley, A.; Hamburg, M. Post-quantum cryptography in TLS 1.3. In Proceedings of the Network and Distributed Systems Security (NDSS) Symposium, San Diego, CA, USA, 24–27 February 2019.
37. Crockett, E.; Paquin, C.; Stebila, D. A Scalability Assessment of Post-Quantum TLS in Cloud Environments. *IACR Cryptol. ePrint Arch.* **2023**, *2023*, 318.
38. Drucker, N.; Gueron, S. Performance Analysis of Post-quantum TLS 1.3 on Constrained Devices. In Proceedings of the Selected Areas in Cryptography—SAC 2021, Taipei, Taiwan, 23–25 November 2022; Springer: Cham, Switzerland, 2022; pp. 427–446.
39. Fouque, P.A.; Tunstall, M. Side-Channel Attacks on Post-Quantum Cryptographic Schemes. *IACR Trans. Cryptogr. Hardw. Embed. Syst.* **2022**, *2022*, 1–45.
40. Xu, Z.; Pemberton, O.; Achutha, R. A Systematic Evaluation of the Security of Post-Quantum Cryptography Against Side-Channel Attacks. In Proceedings of the Cryptographic Hardware and Embedded Systems—CHES 2022, Leuven, Belgium, 18–21 September 2022; Springer: Berlin/Heidelberg, Germany, 2022; pp. 220–244.
41. Basu, K.; Soni, D.; Nabeel, M.; Karri, R. FPGA Acceleration of Post-Quantum Cryptographic Algorithms. In Proceedings of the Design, Automation & Test in Europe Conference (DATE), Virtual, 1–5 February 2021; IEEE: Piscataway, NJ, USA, 2021; pp. 1506–1511.

42. Banerjee, U.; Pathak, A.; Chandrakasan, A.P. Accelerating Post-Quantum Cryptography Through Specialized Hardware for Integer Polynomial Multiplication. In Proceedings of the IEEE International Symposium on Circuits and Systems (ISCAS), Austin, TX, USA, 27 May–1 June 2022; IEEE: Piscataway, NJ, USA, 2022; pp. 1–5.
43. Safe, O.Q. liboqs: C Library for Quantum-resistant Cryptographic Algorithms. 2023. Available online: https://github.com/open-quantum-safe/liboqs (accessed on 25 March 2025).

Disclaimer/Publisher's Note: The statements, opinions and data contained in all publications are solely those of the individual author(s) and contributor(s) and not of MDPI and/or the editor(s). MDPI and/or the editor(s) disclaim responsibility for any injury to people or property resulting from any ideas, methods, instructions or products referred to in the content.